Doc Code: AP.PRE.REQ

# PRE-APPEAL BRIEF REQUEST FOR REVIEW

| Docket Number (Optional) |
|---|
| 42P9141 |

| I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to "Mail Stop AF, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450" [37 CFR 1.8(a)] | Application Number | Filed |
|---|---|---|
| | 09/802,787 | March 8, 2001 |
| on ___October 5, 2005___ Signature_____ | First Named Inventor Khanna et al. | |
| Typed or printed name ___Adrian Villarreal___ | Art Unit 2128 | Examiner Day, Herng Der |

Applicant requests review of the final rejection in the above-identified application. No amendments are being filed with this request.

This request is being filed with a notice of appeal.

The review is requested for the reason(s) stated on the attached sheet(s).
Note: No more than five (5) pages may be provided.

I am the

☐ applicant/inventor.

☐ assignee of record of the entire interest.
See 37 CFR 3.71. Statement under 37 CFR 3.73(b) is enclosed.
(Form PTO/SB/96)

☒ attorney or agent of record.
Registration number ___46,149___

☐ attorney or agent acting under 37 CFR 1.34.

Registration number if acting under 37 CFR 1.34 _____

_____
Signature

___R. Alan Burnett___
Typed or printed name

___(206) 292-8600___
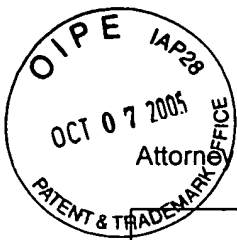Telephone number

___Oct 5, 2005___
Date

NOTE: Signatures of all the inventors or assignees of record of the entire interest or their representative(s) are required. Submit multiple forms if more than one signature is required, see below*.

☐ *Total of _____ forms are submitted.

Attorney Docket No. 42P9141

*Patent*

## IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of: )
)
    Khanna et al. )  Examiner: Day, Herng Der
)
Serial No. 09/802,787 )  Art Unit: 2128
)
Filed: March 8, 2001 )
)
For:   METHOD FOR REPRESENTING ROOT)
        BUSSES USING OBJECT ORIENTED )
        ABSTRACTIONS )

Mail Stop AF
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

## PRE-APPEAL BRIEF REQUEST FOR REVIEW

Sir:

In response to the Final Office Action mailed July 5, 2005 and the Advisory Action mailed September 22, 2005, and in concurrence with the Applicants' Notice of Appeal filed herewith, please consider the following remarks.

## REMARKS

In the Final Office Action dated July 5, 2005, the Examiner rejected claims 1-24 under 35 U.S.C. § 103(a) as being unpatentable over Furner *et al.*, U.S. Patent No. 5,974,474 (hereinafter *Furner*), in view of Dinallo, U.S. Patent No. 5,727,212 (hereinafter *Dinallo*). Applicants respectfully assert the rejections are clearly in error.

Claim 1 is illustrative of the claimed invention, reciting:

1. A method for representing a root bus of a computer system, comprising:

*dynamically* generating an object-oriented abstraction corresponding to the root bus referencing one or more methods that may be implemented to obtain and/or generate configuration and resource allocation information *for the root bus and any subordinate busses connected either directly or indirectly to the root bus*; and

*registering the methods* referenced in the object-oriented abstraction via a data structure stored in memory of the computer system. (Emphasis Added)

In support of the § 103(a) rejection of claim 1, the Examiner states,

> Regarding claim 1, Furner et al. disclose a method for representing a root bus, comprising:

> dynamically [generating an object-oriented abstraction corresponding to the root bus referencing one or more methods that may be implemented] to obtain and/or generate configuration and resource allocation information for the root bus and any subordinate busses connected either directly or indirectly to the root bus (configuration process includes resolving conflicts, FIG. 13); and

> Furner et al. also disclose peripheral buses are characterized by being connected, directly or indirectly, to the CPU-memory bus through bus controllers that actively manage the communication to the hardware devices on the bus (column 10, lines 13-16). Furthermore, Furner et al. disclose an installation information table as shown in FIG. 2E, which implies registering functions. However, Furner et al. fail to expressly disclose defining an object-oriented abstraction including methods. Nevertheless, Furner et al. do suggest the abstracting into functional modules. Thus, applications could refer to hardware instances in a common manner (column 31, lines 41-51).

> Dinallo discloses bridging communication between an object oriented component and a procedural programmed device driver (Dinallo, column 2, lines 4-16). As shown in Fig. 3, Object includes multiple methods to provide different functions. In the OOP environment, registering the methods is well known.

> It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teachings of Furner et al. to

incorporate the teachings of Dinallo to obtain the invention as specified in claim 1 because bridging communication between an object oriented component and a procedural programmed device driver the existing procedural programmed device driver could be reused in the OOP environment.

*Furner* uses an identification and configuration system 119 to identify hardware devices and hardware instances of those devices. With respect to the Examiner's comment of, "Furner et al. disclose an installation information table as shown in FIG. 2E, which implies registering functions," the installation information table stores information related to hardware device drivers. In further detail, *Furner* states,

> As will be explained in detail below, the identification and configuration system 119 of the present invention determines the optimal driver 121 for a particular hardware instance 150 by comparing various characteristics of all the drivers that are capable of supporting the hardware instance. To determine which drivers 121 can support a particular hardware instance, the identification and configuration system 119 compares the information in each driver record 241 with that in the hardware device records 240. *This process identifies drivers 121 that can support the particular hardware instance 150.* (Col 15, lines 8-18, emphasis added)

and

> Once an optimal *driver* 121 is selected, the identification and configuration system 119 places installation information for the selected *driver* into an *installation information table* 133 shown in FIG. 2E. The installation table 133 includes such information as the driver name 207, driver location 208, resource settings 142, and HIN number 222. *The identification and configuration system 119 uses the installation information table 133 to configure the hardware instance 150 in a manner described below.* (Col 15, lines 46-54, emphasis added)

It is clear from above that the installation information table 133 stores information identifying a selected driver for a corresponding hardware instance (*e.g.*, a hardware device driver). Such a driver is employed to access and configure its corresponding hardware instance. *Furner* does not store any information relating to "one or more *methods* that may be implemented to obtain and/or generate configuration and resource allocation information *for the <u>root bus</u> and any <u>subordinate busses</u> connected either directly or indirectly to the <u>root bus</u>*. Under *Furner*, configuration and resource allocation information for root busses and subordinate busses is obtained by the identification and configuration system 119. There is no <u>dynamic</u> identification of a method or methods to perform this operation, or registration of such methods in memory – the process and

method are known *a priori* (in advance). The only functions that are registered under *Furner* relate to drivers specific to respective hardware devices and/or hardware instances.

The Examiner further makes reference to Col. 31, lines 40-51, which states,

> There are portions of this identification and configuration system 119 which could be abstracted into functional modules, as in a functional programming language or object-oriented programming language such as C++, such as the automatic identification mechanism 301 and the configuration mechanism 302. These functional modules could be made available to applications such as installation utilities as shown above, monitoring software and network management software *for the purpose of identifying and configuring hardware instances*. Thus, applications could refer to hardware instances in a common manner. (Emphasis added)

The foregoing explicitly states that the functional modules could be employed *for the purpose of identifying and configuring hardware instances*. The one or more methods recited in claim 1 concern methods that are use to obtain and/or generate configuration and resource allocation information *for the root bus and any subordinate busses connected either directly or indirectly to the root bus*. Clearly, the methods referred to in claim 1 serve an entirely different function than the device drivers employed by *Furner* (which conceivably could be accessed via the theoretical functional modules discussed above).

With respect to this last argument, which was also made in the Response mailed by Applicants on March 25, 2005, the Examiner states in paragraph 9-2,

> Response to Applicants' arguments (2) and (3). By way of identifying and configuring for hardware instances, configuration and resource allocation information are generated at each associated table. For example, a number of resource settings 142 allocate system resources to a hardware instance 150 (column 12, lines 19-51). In other words, the disclosure of Furner et al. is not drivers only as asserted by the Applicants.

Whether or not Furner discloses configuration information for drivers only is not the issue here. The issue from above that was not addressed by the Examiner's response is whether Furner discloses the use of anything that identifies one or more methods that may be implemented to obtain and/or generate configuration and resource allocation information *for the root bus and any subordinate busses connected either directly or indirectly to the root bus*. Clearly Furner does not teach or suggest this limitation.

In view of the foregoing argument it is clear that the combination of *Furner* and *Dinallo* do not teach or suggest all of the elements and limitations recited in claim 1, as

required by the third prong of the *In Re Vaeck* test. Accordingly, the rejection of claim 1, as being unpatentable over *Furner* in view of *Dinallo* is unsupported and thus improper and should be withdrawn.

With respect to independent claims 9 and 19, each of these claims recite elements similar to claim 1 with respect to the use of an object oriented representation of root busses. Accordingly, independent claims 9 and 19 are patentable over the combination of *Furner* and *Dinallo* for at least the same reasons as claim 1.

If a telephone interview would expedite the prosecution of this application, the Examiner is invited to contact Alan Burnett at (206) 292-8609.

If there are any additional charges/credits, please charge/credit our deposit account no. 02-2666.

Respectfully submitted,

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN

Date: *Oct 5, 2005*                          R. Alan Burnett
Reg. No. 46,149

12400 Wilshire Boulevard
Seventh Floor
Los Angeles, CA 90025-1030